WHAT IS CLAIMED IS:

1.   A method of building a combined workflow comprising:

accepting a first workflow comprising a first plurality of tasks and associated with a first party;

accepting a second workflow comprising a second plurality of tasks and associated with a second party;

ordering the first plurality of tasks and the second plurality of tasks into a combined workflow having a task order that, when executed, provides a desired result of a business collaboration between the first party and the second party; and

adding ordering tasks operable to implement the order of the combined workflow and thereby achieve the desired result.

2.   The method of claim 1 wherein adding ordering tasks comprises forming a sequential flow which interleaves implementation of the first plurality of tasks and the second plurality of tasks.

3.   The method of claim 1 wherein adding ordering tasks comprises forming a parallel flow of a first task within the first plurality of tasks and a second task within the second plurality of tasks.

4.   The method of claim 1 wherein adding ordering tasks comprises adding at least one of conjunctive splitting and joining tasks which specify the task order.

5.   The method of claim 1 wherein adding ordering tasks comprises adding at least one of alternative splitting and joining tasks which specify the task order.

6.   The method of claim 1 wherein adding ordering tasks comprises adding a first splitting task which designates that a first task within the first workflow is followed by a first following task and a second following task.

7.   The method of claim 6 wherein adding ordering tasks comprises adding the first following task as a second task within the second workflow.

8.   The method of claim 6 wherein adding ordering tasks comprises adding the first following task as a first joining task, the first joining task designating a second task within the second workflow as following the first joining task and the first splitting task.

9.   The method of claim 8 wherein adding ordering tasks comprises adding a second splitting task following the second task within the second workflow, the second splitting task designating that the second task is followed by a third following task and a fourth following task.

10.   The method of claim 9 wherein adding ordering tasks comprises adding the third following task as the second following task, the second following task being a second joining task within the first workflow that designates that a third task within the first workflow follows the second following task.

11.   The method of claim 10 wherein adding ordering tasks comprises adding the fourth following task as a third joining task within the second workflow, the third joining task designating that a fourth task within the second workflow follows the third joining task and the third task within the first workflow.

12.   The method of claim 11 wherein a second ordering task is a joining task which designates that a fourth task within the second workflow, the fourth task following the second task within the combined workflow.

13.   The method of claim 9 wherein adding ordering tasks comprises:
       adding a third task within the first workflow as the second following task;
       adding a second joining task within the first workflow as the third following task, the second joining task designating that a fourth task within the first workflow follows the third following task.

14. The method of claim 1 wherein ordering the first plurality of tasks comprises inputting the task order from an operator.

15. The method of claim 14 wherein adding ordering tasks comprises:

representing the first workflow as a first matrix in which the first plurality of tasks are each represented as first vertices, where values of the first vertices within the first matrix are determined by first dependencies between the first plurality of tasks; and

representing the second workflow as a second matrix wherein each of the second plurality of tasks are represented as second vertices, where values of the second vertices within the second matrix are determined by second dependencies between the second plurality of tasks.

16. The method of claim 15 wherein adding ordering tasks comprises:

inserting the first matrix and the second matrix into a third matrix;

modifying a selected value within the third matrix, thereby reflecting a construction or removal of a selected dependency between two vertices within the first plurality of tasks, consistent with the task order;

adding a fourth vertex before a first of the two vertices, the fourth vertex having a first chosen value reflecting a first new dependency between the fourth vertex and the first of the two vertices; and

adding a fifth vertex after the first of the two vertices, the fifth vertex having a second chosen value reflecting a second new dependency between the fifth vertex and the first of the two vertices.

17. The method of claim 1 wherein the first workflow is an abstracted workflow associated with a first actual workflow of the first party, and further wherein a confidential nature of the first actual workflow is protected by use of the abstracted workflow in constructing the combined workflow.

18.   The method of claim 1 further comprising selecting a subset of the combined workflow for execution by the first party.

19.   The method of claim 18 wherein selecting a subset comprises determining that the subset includes a third plurality of tasks, each consecutive pair of the third plurality of tasks connected by a dependency.

20.   The method of claim 18 wherein selecting a subset comprises determining that a last task within the third plurality of tasks precedes at most one subsequent task within the combined workflow.

21.   The method of claim 20 wherein selecting a subset further comprises determining that no internal task within the third plurality of tasks, exclusive of the last task, immediately precedes an external task that is not included within the third plurality of tasks.

22.   The method of claim 20 wherein selecting a subset further comprises determining that no internal task within the third plurality of tasks, exclusive of a first task of the third plurality of tasks, immediately succeeds an external task that is not included within the third plurality of tasks.

23.   An apparatus comprising a storage medium having instructions stored thereon, the instructions including:

a first code segment for accepting a first workflow comprising a first plurality of tasks and associated with a first party;

a second code segment for accepting a second workflow comprising a second plurality of tasks and associated with a second party;

a third code segment for accepting a task order for forming the first plurality of tasks and the second plurality of tasks into a combined workflow, wherein the combined workflow, when executed, provides a desired result of a business collaboration between the first party and the second party; and

a fourth code segment for adding ordering tasks operable to implement the order of the combined workflow and thereby achieve the desired result.

24. The apparatus of claim 23 wherein the fourth code segment comprises a fifth code segment for forming a sequential flow which interleaves implementation of the first plurality of tasks and the second plurality of tasks.

25. The apparatus of claim 23 wherein the fourth code segment comprises a fifth code segment for forming a parallel flow of a first task within the first plurality of tasks and a second task within the second plurality of tasks.

26. The apparatus of claim 23 wherein the fourth code segment comprises a fifth code segment for adding at least one of conjunctive splitting and joining tasks which specify the task order.

27. The apparatus of claim 23 wherein the fourth code segment comprises a fifth code segment for adding at least one of alternative splitting and joining tasks which specify the task order.

28. The apparatus of claim 23 wherein the third code segment comprises a fifth code segment for inputting the task order from an operator.

29. The apparatus of claim 28 wherein the fourth code segment comprises:

a fifth code segment for representing the first workflow as a first matrix in which the first plurality of tasks are each represented as first vertices, where values of the first vertices within the first matrix are determined by first dependencies between the first plurality of tasks; and

a sixth code segment for representing the second workflow as a second matrix wherein each of the second plurality of tasks are represented as second vertices, where values of the second vertices within the second matrix are determined by second dependencies between the second plurality of tasks.

30. The apparatus of claim 29 wherein the fourth code segment comprises:

a seventh code segment for inserting the first matrix and the second matrix into a third matrix;

5   an eighth code segment for modifying a selected value within the third matrix, thereby reflecting a construction or removal of a selected dependency between two vertices within the first plurality of tasks, consistent with the task order;

a ninth code segment for adding a fourth vertex before a first of the two vertices, the fourth vertex having a first chosen value reflecting a first new dependency between the fourth 10   vertex and the first of the two vertices; and

a tenth code segment for adding a fifth vertex after the first of the two vertices, the fifth vertex having a second chosen value reflecting a second new dependency between the fifth vertex and the first of the two vertices.

15   31. The apparatus of claim 23 wherein the first workflow is an abstracted workflow associated with a first actual workflow of the first party, and further wherein a confidential nature of the first actual workflow is protected by use of the abstracted workflow in constructing the combined workflow.

20   32. The method of claim 23 further comprising a fifth code segment for selecting a subset of the combined workflow for execution by the first party.

33. The method of claim 32 wherein the fifth code segment comprises a sixth code segment for determining that the subset includes a third plurality of tasks, each consecutive 25   pair of the third plurality of tasks connected by a dependency.

34. The method of claim 32 wherein the fifth code segment comprises a sixth code segment for determining that a last task within the third plurality of tasks precedes at most one subsequent task within the combined workflow.

30

35.   The method of claim 34 wherein the sixth code segment comprises a seventh code segment for determining that no internal task within the third plurality of tasks, exclusive of the last task, immediately precedes an external task that is not included within the third plurality of tasks.

5

36.   The method of claim 34 wherein the sixth code segment comprises a seventh code segment for determining that no internal task within the third plurality of tasks, exclusive of a first task of the third plurality of tasks, immediately succeeds an external task that is not included within the third plurality of tasks.

10